170-WP-001-001

IAS Evaluation Document

White Paper

March 1995

Prepared Under Contract NAS5-60000

White Paper--Not intended for formal review or government approval.

RESPONSIBLE ENGINEER

Judy Feldman /s/ for	3/29/95
Son Dao	Date

EOSDIS Core System Project, Hughes Research Lab

SUBMITTED BY

Mark Settle	3/29/95
Mark Settle, Deputy Project Manager	Date

EOSDIS Core System Project

Hughes Applied Information Systems
Landover, Maryland

This page intentionally left blank.

Abstract

This white paper summarizes the Independent Architecture Studies performed under the EOSDIS Core System (ECS) Contract. An independent review was performed by Hughes Research Laboratory let by Son Dao. On September 21-23, 1994, a panel met to review the results of the EOSDIS Independent Architecture Studies. This report provides ECS independent consensus findings and is intended to document the independent review, to provide guidance to ECS, and to provide feedback to those who performed the studies.

The three studies were performed by teams from the University of California, George Mason University together with scientists from the Universities of Delaware and New Hampshire, and the Universities of North Dakota and Nebraska. Their three studies differed considerably in what they offered but were complementary. Each study focused on a different segment of the user community; the UC team focused on Earth science researchers with strong focus on particular sites and considerable abilities to reinterpret remote sensing, the North Dakota team focused on applications users such as those providing data to the agriculture community, and the George Mason team took a less focused view, but included Earth science researchers as well as those who would use the data for other government purposes. One major conclusion of our panel was that the architecture one proposes is strongly influenced by ones view of who are the users.

Key Words: *Architecture, Independent Architecture, Database, User Model, Network Management, Information Management*

This page intentionally left blank.

Contents

This page intentionally left blank.

1. Introduction

1.1 Purpose

This white paper summarizes the Independent Architecture Studies performed under the EOSDIS Core System (ECS) Contract. An independent review was performed by Hughes Research Laboratory let by Son Dao. On September 21-23, 1994, a panel met to review the results of the EOSDIS Independent Architecture Studies. This report provides ECS independent consensus findings and is intended to document the independent review, to provide guidance to ECS, and to provide feedback to those who performed the studies.

1.2 Organization

This paper is organized as follows: Section 2 provides a high level summary of the IAS. Section 3 presents the IAS evaluation process. Section 4 details each of the three IAS study recommendations. The members of the review panel are listed in Appendix A. Appendix B contains the IAS summary charts presented during the SDPS Preliminary Design Review in February, 1995.

1.3 Review and Approval

This White Paper is an informal document approved at the Office Manager level. It does not require formal Government review or approval; however, it is submitted with the intent that review and comments will be forthcoming.

Questions regarding technical information contained within this Paper should be addressed to the following ECS and/or GSFC contacts:

- ECS Contacts
  - Mark Settle, Deputy Project Manager, (301) 925-0484, msettle@eos.hitc.com
- GSFC Contacts
  - Gail McConaughy, Systems Engineer, (301) 286-7741, gmcconaughy@gsfc.nasa.gov

Questions concerning distribution or control of this document should be addressed to:

Data Management Office
The ECS Project Office
Hughes Applied Information Systems
1616 McCormick Dr.
Landover, MD 20785

This page intentionally left blank.

2. Summary of the Review Panel for the EOSDIS IAS

On September 21-23, 1994, a panel consisting of those listed in Appendix A met to review the results of the EOSDIS Independent Architecture Studies. This report provides the HRL consensus findings and is intended to document HRL review, to provide guidance to Hughes, and to provide feedback to those who performed the studies.

The three studies were performed by teams from the University of California, George Mason University together with scientists from the Universities of Delaware and New Hampshire, and the Universities of North Dakota and Nebraska. Their three studies differed considerably in what they offered but were complementary. Each study focused on a different segment of the user community; the UC team focused on Earth science researchers with strong focus on particular sites and considerable abilities to reinterpret remote sensing, the North Dakota team focused on applications users such as those providing data to the agriculture community, and the George Mason team took a less focused view, but included Earth science researchers as well as those who would use the data for other government purposes. One major conclusion of our panel was that the architecture one proposes is strongly influenced by ones view of who are the users.

In summary, the California team proposed an alternative architecture featuring a data base centric view of EOSDIS. Their design solution featured heavy centralization of data at two super DAACs along with a less clear set of extensions to "Peer DAACs" whose functions were to provide more products than the super DAACs but with less reliability. The study provided creative new names for a number of new and existing concepts and system elements and generally provided a crisp and clear analysis of functions from a computer science standpoint. The study's database centric point of view clearly reflected the considerable excellence of the team in the world of data base systems. The study acknowledged that current DBMS systems will need considerable extensions in order to meet EOSDIS requirements but felt that with specific funding targeted to medium and small sized DBMS and Hierarchical File Management System (HFMS) COTS vendors, products would come to market which would meet EOSDIS' needs in a timely fashion. The team acknowledged considerable ties to DBMS companies. In general, the issues of performance were addressed by advocating the acquisition of more hardware (or iron as the study referred to it) to achieve whatever was desired. The study emphasized the need to invest considerable resources in database design efforts. In particular, the concept of queries from hell was discussed and it was acknowledged that such massive queries of the entire data base would be accomplished as part of roughly monthly cycles of reading/refreshing the database.

The North Dakota - Nebraska team devoted their study to definition of the concept of a PARC. PARCs would be extensions of EOSDIS to focus on specific user communities such as the agriculture community in a particular region. PARCs would provide data products tailored to the needs of such communities. The team advocated that PARCs start as government sponsored entities but acknowledged that they could evolve into for profit, value added businesses. The study highlighted the need for timely availability of data, pointing out that many data are degraded in value with time. They strongly supported making data available through direct broadcast from EOS satellites. In general, the key architectural issue raised for EOSDIS is the need to ensure that extensions to the system can be implemented and that data from EOS can be made available in a timely fashion.

The team led by George Mason University (GMU) provided a specific architectural design that was generally compatible with the overall institutional building blocks and division of responsibilities of the baseline EOSDIS architecture. The point of view adopted by the team could be characterized as Global Change DIS or Environmental studies DIS. The architecture assigns different types of information, such as levels of metadata, to different servers. The concept of a Virtual Client Protocol was advocated to enable servers responding to queries relayed through other servers to provide responses directly to the original requesting client. This study provided a wealth of detailed ideas and analyses which should prove useful to Hughes system implementers. In particular, the user scenarios provided by this study should be added to the set of scenarios currently being used by Hughes. This study also pointed out the need for connection to many external data systems both to extend EOSDIS and to provide user access to additional data.

The Panel concluded that there need to be many extensions of EOSDIS to better serve specific user groups, to provide enhanced focus on select products, and to off-load the DAACs and that EOSDIS must maximize its ability to adapt and evolve. The EOSDIS problem has not been solved before, and its solution requires ideas and concepts which have not been instantiated. There is no change to the current approach for EOSDIS implementation which would greatly improve the system while still meeting the current requirements—there is no magic bullet. A number of the recommendations from the alternative architecture studies dealt with changing the requirements on EOSDIS. These go beyond assessing alternative architectures, but the issues raised should be examined by NASA to ensure that maintaining these requirements will not preclude critical efficiencies in EOSDIS implementation. The studies have provided a number of detailed ideas to be examined on all time scales whose implementation may improve EOSDIS. We spent considerable time discussing these ideas and the options they present system development. Our assessment of these items and the time scale on which they should be considered is described in Chapter 3.

The alternative architecture studies, particularly that of the California team, raised a significant question as to whether the EOSDIS baseline approach is data base centric. By this term we mean the degree to which the system architecture and design will use the considerable power of data base management systems to accomplish its functions. Our panel included members who felt strongly that this was the only way to go as well as others who expressed reservations. Hughes provided a succinct exposition of the way in which data base management systems are included in the current architecture and their expectations as to how they would factor into system design. Our conclusion, based on this explanation, is that the design is currently data base centric to a degree commensurate with the expected readiness of extensions to current data base management systems and that the design does not preclude an evolution to a full data base centric approach. Furthermore, we concluded that this represents a reasonable approach for now.

All three study teams raised significant questions as to whether the Common Object Request Broker Architecture (CORBA) would be ready to meet the needs of EOSDIS in the future. The current baseline architecture assumes an eventual evolution to CORBA. This cautionary note should not be forgotten.

There are six items which we would like to emphasize specifically which merit further study or effort.

1. The California team proposed two identical archives each of which would hold all the data. This points to potential savings in having a common system design for all archiving elements of EOSDIS. Thus, a study should be made to determine if EOSDIS would be benefited by having the same archive system design applied to the Level 0 back-up archive as is being used for the DAACs.
2. EOSDIS should examine its various software development plans to determine if there are cases in which requirements could be better met by spending money to influence COTS development. Hierarchical Storage Management Systems may present one such area for application of this development technique.
3. At every step, the architecture of EOSDIS should be analyzed and tested to ensure that it easily enables extensions to the system by others.
4. The User Model must be clearly established and related to the services that can be provided within overall cost constraints on the system.
5. The current EOSDIS design permits support for bringing user defined processes or functions to the data centers to be run. In essence, this is bringing the process to the data. The current implementation plan does not provide computational resources at the data centers to support this. The question of bringing the process to the data versus bringing the data to the process should be reexamined and, if possible, a means should be found to allow this decision to be made adaptively.
6. There are considerable questions surrounding the issue of routine production of standard products. While the California team pointed to users who would need only the lowest level of standard product, all teams pointed to the need to produce products tailored to specific sets of users. The decision as to what products to produce routinely is clearly a function of the actual user needs as well as one of capturing measurement related expertise which may otherwise be lost over time. The routine production decision is also entangled with the costs the user will pay under the principle of charging no more than the cost of fulfilling the user request (COFUR). If a product is routinely produced the user bears no cost for its production, but if it is processed on demand the costs of production may be charged to the user. This situation requires further study to determine its effects on the efficient operation of the system.

For those members of the panel with long EOSDIS experience, the review of these studies pointed up a significant need to better communicate the key concepts of EOSDIS and the approach being taken in its development. There were so many critical instances where the study teams misunderstood some aspect of the current system approach or some principle governing its requirements that fault must lie in the communication by NASA and its contractors about EOSDIS. The panel strongly recommends that this situation be remedied.

In conclusion, the panel acknowledges the considerable work and inventiveness demonstrated by the independent architecture study teams. Although each of the studies took a different approach to their work and delivered different products, EOSDIS should benefit from the work of all three teams. The panel thanks the personnel at Hughes for their excellent support to us in conducting this review.

3. ECS Independent Architecture Studies Evaluation Process

In performing our analysis of the independent architecture studies, we have extracted key recommendations made by the teams and have summarized the potential impact of these recommendations on the ECS Program. We found that the clearest way to characterize these recommendations is to label them according the following five major groups:

a) ECS –Items in this category are currently under full or partial consideration by the HAIS team, or are currently incorporated into the HAIS architecture. The detailed approach or implementation used in the HAIS architecture may vary from the approach suggested in the university study, however the primary functionality remains comparable.

b) PDR (Preliminary Design Review)–Items in this category are of immediate relevance to the PDR and should be assessed as implementation solutions at the PDR. Whether or not the item is currently under consideration by HAIS, the particulars of the suggested implementation warrant specific attention from the HAIS team.

c) EVO (Evolvability)–Items in this category should be considered for future generations of the ECS architecture. These are items that the current architecture must eventually be able to support through a reasonable evolutionary path. We have further divided this category into both short-term (S) and long-term (L) classifications. The ECS architecture must be able to support short-term items within two to three years, and it must be able to support long-term items within three to five years. The long-term items are considered insufficiently mature, and need to be reassessed at a later time. In some cases, items in this category are marked both short and long term. This is done for suggestions that have elements that fall into both classifications.

d) R&D (Research)–Items in this category are considered worth investigation, but their potential cannot yet be realized without further research. This category is applied to unproven technologies and promising concepts. Further research in these areas is likely to provide benefit to the ECS program.

e) PRG (Programmatic)–Items in this category may require further review at the programmatic level. In most cases, they are recommendations which are outside the scope of the current ECS program, or they would require reorganization of the program structure in order to be implemented.

In many instances, individual suggestions from the university teams were placed in more than one of the above groups. This is because these groups are not mutually exclusive. For example, a single suggestion may be seen both as needing assessment for the PDR while a similar concept might already be under consideration by the ECS team. Likewise, a concept may be seen as important for the future evolvability of the system while it also has certain elements that require further research.

This page intentionally left blank.

4. IAS Team Results

4.1 Summary and Analysis of the UCB Independent Architecture Study

The U.C. Berkeley team presents a "database centric" architecture for EOSDIS. Based on the analysis of several user scenarios, the team concludes that a completely flexible approach to the "standard production" vs. “compute-on-demand” strategy is required for all processing within EOSDIS. This approach is necessitated by the evolutionary nature of the production algorithms, the desire of high-end users to have products tailored to their individual research needs, and the unpredictable nature of product demands.

To support this strategy, the team presents a “database centric” architecture that advocates support for advances in Object-Oriented Relational DataBase (OORDB) technology which will form a substantial portion of the proposed software architecture for the Science Data Processing Segment. SQL-* (current vendor implementations of abstract data types and distributed query language) is proposed as the standard query language that will be used to support the proposed middleware. Location and data distribution transparencies will be supported by the middleware. Database triggers and views are employed to support a dynamic "eager vs. lazy" evaluation production strategy that can respond to changing needs on a product-by-product basis. The team also proposes an incremental implementation of a rich Earth Science type library using abstract data type mechanisms. An alliance with one or more commercial database vendors and HSM vendors is proposed as the mechanism to deliver a tertiary-storage-backed distributed database engine.

The team proposes a system design based on a "2 + N" model that reallocates the functionality of EOSDIS across 2 superDAACs, and N peerDAACs, with the superDAACS responsible for processing and archiving level 1 and (most of) level 2 data, and the peerDAACs responsible for generating and distributing the remainder. The team suggests that true archival of the data only be performed for level 1, with the remainder of the data being reproducible on demand. The architecture employs a distributed approach to both computing and storage, based on the team's analysis of technology trends and price-performance curves that favor large numbers of distributed centers employing commodity technologies over a smaller number of centers employing higher end technologies.

The team identifies potential risks associated with the proposed software and hardware architecture, and develops a cost analysis based on the proposed functionality and their assessment of technology trends. The team also includes an initial design of a representative database schema for the scenarios presented, and illustrates the use of queries in accessing and manipulating the data.

In the following sections, we identify and analyze the potential impact of the team’s major ideas and suggestions to the ECS program using the previously described evaluation classification.

The team technical contribution is categorized into three areas: information management, network management, and other architecture issues.

4.1.1 Information Management

- **Rearchitect DAACs and SCFs**

  Categorization: PRG

The UCB team suggests redistributing the current DAAC and SCF functionality in a “2 superDAACs + N peerDAACs” model. In addition to storing the entire collection of raw data, the superDAACs will precompute the results to commonly requested queries and perform processing required for some ad hoc queries. The additional N peerDAACs will respond to ad hoc queries and store data sets commonly requested by a particular subset of the Earth Science community. The peerDAACs will be fashioned from commodity computing and storage technologies and have no formal performance and/or reliability requirements, making them smaller and cheaper than the superDAACs.

The redistribution of functionality as suggested has severe programmatic restructuring implications for EOSDIS. However, it should be noted that the current ECS architecture does not depend on any particular configuration of "DAACs" and "SCFs". Rather, it allows an arbitrary number and configuration of "providers," with the actual roles dictated by policy. Support for service reconfiguration and migration is one of the features integrated into the architecture as a result of the System Requirements Review and as demonstrated at the System Design Review.

- **Focus on ad hoc queries and flexible strategy for computing standard products**

  Categorization: PDR, EVO (S/L), PRG

The premise of the UCB team is that there will be few true "standard" products over the course of time. Rather, they believe that users will gravitate towards a more dynamic interaction model, in which desired data is discovered, produced and requested through an ad hoc querying process, as opposed to more structured requests for standard products. Their proposed architecture reflects this belief, providing all processing (both standard products and ad hoc requests) through a common mechanism. This mechanism allows optimization between precomputed "standard" products, and compute-on-demand "ad hoc" requests at all levels of processing, responding dynamically to user requests and the availability of computational/storage resources.

Support for a dynamic compute vs. store tradeoff on a per product basis has been integrated into the system design, again based on SRR feedback. Current consensus among the instrument teams, and the results of the Ad Hoc Working Group on Production, appear to predict a somewhat less dynamic environment than that indicated by the Berkeley team. This environment reflects a fairly tight interdependence of products and some longer term algorithmic stability (after initial fluctuations), leading to more "standard" products than assumed by the UCB team.

Additionally, many of the instrument team algorithms are based on legacy instrument algorithms. The rearchitecting of these algorithms to support a database-centric environment (where inputs are obtained from, and outputs generated into the database) could provide additional flexibility in supporting lazy vs. eager evaluation. However, this rearchitecting comes at some cost, and it is unclear to what extent dynamic lazy vs. eager evaluation can be used in the highly interdependent processing scenarios.

The current production architecture could support a scenario more biased towards lazy evaluation by reallocating computational resources to ad hoc queries. The details of these features in supporting a dynamic lazy vs. eager evaluation optimization will be addressed at PDR in consultation with the Ad Hoc Working Group on Production. Longer term technology evolution in resource management and dynamic scheduling should also be tracked.

- **Different processing modes**

Two major processing modes are proposed by the ~~S2K~~ UCB team.

- **Periodic Cycling of the store ("queries from hell")**

Categorization: PDR, EVO (S/L)

These queries typically require access to large portions of the storage repository, e.g., to "reprocess" an entire dataset using a revised algorithm. Such queries may run for days, or even weeks. The UCB team suggests "batching" such queries periodically, so that a single sequential pass of the stores may support a number of such requests. Techniques are required to recognize this type of query, and the middleware must support collecting together, executing, and disassembling the appropriate answer for such batches.

This was generally received as an excellent suggestion, and is applicable to the current ECS design as well as the UCB architecture. It should be addressed by the ECS team at PDR. Longer term technology evolution should also be employed to support better a priori identification of resource intensive queries, and to provide effective resource scheduling over the extent of potentially long duration queries.

- **Eager vs. lazy evaluation**

Categorization: ECS, PDR, EVO (L), R&D, PRG

The output of every node can be evaluated eagerly when inputs become available; or it can be evaluated lazily when its output is needed. The team believes that all "products" within EOSDIS need to be considered for this optimization, allowing dynamic distribution of resources to the push (eager) and a pull (lazy) sides of the system as motivated by user demand. The optimization cost model and algorithm to support such a dynamic resource allocation problem is very complex, and further research is needed. In addition, queuing and workflow management techniques for organizing and scheduling triggers and views require plans for long term evolvability and R&D effort. A short description of the cost model and an example of using database triggers and views to support this evaluation are provided in Section 3 of the report.

This item is closely related to the discussion "Focus on ad hoc queries and flexible strategy ..." above. The current ECS architecture supports both precompute-and-store, and compute-on-demand generation of ECS "products," as recommended by the science community. There is additional interest in determining how flexible the PDR implementation will be in this regard, and how effective the architecture is in reallocating computational resources in such a scenario. At a programmatic level, there will be a need to establish policy on the lazy vs. eager evaluation question. This may require streamlined decision processes for migrating algorithms between these classes of execution.

- **SQL-* is the principal access language**

  Categorization: ECS, EVO (S/L)

The UCB team suggests the use of SQL-* as the query language to retrieve data, and as the intermediate language for communicating among middleware, target DBMSs, and clients. Appropriate translation to SQL-* is needed to support a variety of client languages and target DBMS languages. And by having SQL-* as the intermediate language, the required number of translations among foreign underlying data servers and clients are reduced from n-square to 2n. The UCB team suggests that database vendors employing SQL-* will be developing a number of such gateways over time in order to integrate legacy and new systems. The features of SQL-* required to support the proposed architecture are described in the sub-bullets below.

The current ECS architecture plans to support a number of protocols, including SQL dialects with type extensions. Consistent with the philosophy of heterogeneity and custom client interfaces, the ECS architecture plans to support multiple protocols, such as HTTP and Z39.50. This leaves service implementation decisions up to the provider, allowing protocols well matched to the provider's capabilities and desired level of service. Because of the volatile nature of most of these standards, the ECS team has been hesitant to "bank" on just one. There are particular concerns with SQL-* in that it represents current vendor implementations of portions of the current SQL3 standard. With the SQL3 feature set unclear, and 3-5 years away from standardization, the ECS team has been hesitant to commit to what is, in essence, an unknown standard. Given the shorter term deliverables (release A in 12/96), reliance on SQL-* and SQL3 alone seems risky. ECS provides an architecture that allows multiple protocols to be supported since there are heterogeneous Earth data systems. This architecture doesn’t exclude the support for SQL-* protocol. Protocol trade studies are underway at the present time and will be reported on at PDR. Longer term, the ECS team will need to follow the SQL3 and OQL (OMG) standards, and will need to support these standards as commercial implementations are realized.

- **Type extension & functions & type library**

  Categorization: ECS, EVO (L), R&D

SQL-* provides the mechanisms required for type extension, but it does not implement the type libraries that will be required for ECS. While a number of more common types will be provided by commercial database vendors (arrays, multimedia extensions, etc.), it is not clear to what extent ECS-specific data types (e.g., spatial types, time series, event models, etc.) will be implemented. The team suggests that ECS should define the SQL-* type libraries that will extend SQL with EOSDIS-specific data types and functions. The team believes substantial effort is required in this area, since standard libraries for scientific data such as those provided within NetCDF and HDF will need rework to provide the right level of abstraction for a SQL-* representation.

The ECS data model is being developed through Data modeling and Data server design efforts. The model includes Earth Science Data Types (ESDTs), which provide the visible types and services exported by the Data Server, as well as Computer Science Data Types (CSDTs) that are used as implementation vehicles for the ESDTs. This effort provides much of the "functionality" called for by the UCB team, but without the direct linkage back to an extended SQL dialect. The ECS team needs to track the extended relational and OODB implementations to ensure compatibility of its type library with eventual SQL dialects, allowing development of long term functionality in a database-centric environment.

- **Triggers & views for production management**

  Categorization: EVO (L), R&D

The UCB team suggests database *triggers* and *views* as the appropriate mechanisms to support eager and lazy evaluation. Triggers are used to support eager evaluation, allowing arbitrary processing sequences to be initiated in response to events (e.g., receipt of data). Views contain descriptive information pertaining to the underlying data sets, and can be used for lazy evaluation. More R&D efforts are required to advance the concept of triggers and views to the point where they can realistically be used for production management. This includes work on data consistency support for view update, transactional semantics, cross-site triggers, resource management, and process scheduling. Discussions of the usage of triggers, their limitations, and possible extensions for production management are discussed in Section 3.4.2 of the report.

The current ECS approach uses a production management strategy based on graph theoretic algorithms (dependency graphs), resource management strategies, and current process queuing and process management techniques. The UCB team's approach suggests use of trigger and view extensions within the database to accomplish these features. When coupled with processing methods that reside in the database, this would provide a production system that was tightly coupled with the database-based methods, affording the flexibility in lazy vs. eager evaluation the team believes is required. These extensions, particularly in extending transactional semantics of triggers in cross-site execution, and in resource management and process scheduling, are largely research problems at the present time. Hence, ECS should monitor the progress of commercial R&D efforts and preserve the ability to incorporate new technological capabilities over the long term.

- **Access methods & index techniques**

  Categorization: ECS, EVO (L), R&D

The ability to add efficient spatial data retrieval, content-based retrieval, and temporal data access requires the ability to specify complex indexing techniques along with data type extensions. Research is needed for advanced index structuring to efficiently support the retrieval of various spatial data types within ECS.

The ESDTs and CSDTs of the ECS architecture described above must take into consideration appropriate indexing schemes for spatial or value-based (content-based) retrieval. These schemes need to be provided as part of the data type implementations described above. While a number of common spatial indexing schemes exist, other schemes required by ECS may be esoteric, and will require longer term R&D efforts. The ability to support such schemes in the long term needs to be demonstrated as evolutionary criteria.

- **Middleware software**

The UCB team proposes a layer of software that sits between the client and the target DBMSs to support the following features:

- **Location transparency without DCE & CORBA**

  Categorization: ECS, EVO (L)

The UCB team proposes implementation of location and data distribution transparency through the middleware. This middleware provides automatic mapping between a users' location-independent request, and the location-dependent data sources through a distributed query decomposition. The mapping and decomposition are entirely transparent to the user.

The ECS architecture proposes service location transparency through DCE RPC, directory service, and trader functionality, with migration to an OMG base in the longer term. Because the ECS architecture provides service interfaces across a number of protocols, it uses a more general request/reply mechanism on top of a DCE/CORBA foundation. This basic mechanism can be used to "carry" a number higher level "protocols," including extended SQL dialects, HTTP, RPC (DCE IDL), and object messaging (OMG IDL). This is a fundamentally different approach than that taken by the UCB team, but one which provides for location transparent service invocation across a network of heterogeneous providers, a scenario that seems likely in the GCDIS scenario that ECS aims to support. The UCB solution suggests developing a common language (SQL-*) in which all providers agree to converse. The ECS team will need to illustrate adaptability among these various scenarios as a long term evolvability test.

- **Distributed query optimizer for parallelism, load balancing, and copy selection**

  Categorization: ECS, EVO (S/L), R&D

A distributed query optimizer is needed to automatically decompose queries and to select an optimum access plan that will reduce the total execution cost of a query. Based on the data distribution (replicated and fragmented components), the processing load of each site, the network cost, and the complexity of the SQL-* query, the optimizer will attempt to exploit the available parallelism among a subset of the 2+N sites. These issues, as well as that of overall resource management (load balancing) are among of the more challenging research topics being pursued by the database research community.

This is one of the more challenging research efforts required for transparent distributed site search. The ECS architecture supports evolution to a system capable of exploiting advances in these research areas, while minimizing the impact of delays in such technologies. The DIM/LIM hierarchy within ECS will employ distributed query optimization techniques, parallel query execution, copy selection, and load balancing when they are available. However, it is not completely reliant on these advances for successful operation. A number of techniques can be employed within the DIM to provide distinct global "views" of the data to various populations within the ECS community. As distributed query optimization matures, these views can migrate from static representations to more dynamic ones.

- **Should adopt a “DBMS-centric” view point**

The UCB team recommends using an advanced object-relational DBMS as a local storage and processing engine.

- **Instantiate all functions in a DBMS**

  Categorization: EVO (L), R&D

All user-defined functions and user-defined access method should be stored and executed within DBMS boundaries. Maintenance and portability of these functions are still R&D topics.

This approach has a number of advantages, including the ability to simply migrate algorithms from eager to lazy evaluation, and vice-versa. Because all required production functions are housed in the database, they can be invoked in a uniform manner. This approach also provides for simple execution of alternative processing scenarios, as various implementations of data type "methods" can be invoked simply through the SQL-* interface. This approach also allows lineage information to be preserved through simple database operations and system audit trails. However, such an approach requires all algorithms to be developed in a database-centric model, with inputs being "SELECT"ed from the database, and results being "INSERT"ed into the database. For new algorithms and appropriately trained instrument team programmers, this may not be an issue. However, for the substantial legacy instrument code investment, and more traditional "file in - process - fileout" programming models, such an approach requires additional support. The ECS team needs to consider the ramifications of migrating functionality into a database-centric model downstream, as part of its evolution assessment.

- **No file systems besides the one use for DBMSs**

  Categorization: EVO (L), R&D

This is consistent with the UCB team's DBMS-centric approach, and is a laudable goal—providing a single storage abstraction through the database, and handling all "reads" and "writes" to that storage abstraction through database mechanisms. In order to support legacy systems, a software "wrapper" could, in principle, be used to interface file-based algorithms with ECS. However, this type of encapsulation process is very time consuming and has severe limitations depending upon the code structure of the legacy systems. Moreover, ECS performance will degrade significantly.

Longer term, we believe computer system evolution to move in this direction. We are starting to see "database" features in existing operating systems (Macintosh Finder file properties and search, Microsoft's next generation operating systems, etc.). This will tend to blur the distinction between file systems and DBMSs, a trend that ECS will need to plan for in its evolutionary path.

- **Integration of hierarchical storage management (HSM) and DBMS**

  Categorization: EVO (S/L), R&D

The UCB team suggests direct integration of the HSM with the database engine. This provides the required control over caching mechanisms that may not be available through a cached file interface between the DBMS and a file-based HSM. The team suggests substantial investment in advancing the state of this technology, based on its experiences as outlined in Section 3.3 of the report.

The overriding message from the UCB team here is to plan for DBMS systems that are integrated with tertiary storage systems, rather than laid on top of file abstractions on those systems. To that end, they suggest working closely with database and HSM vendors to develop these integrated products. This approach seems advisable even in the more heterogeneous design of ECS, as it will allow ECS to leverage commercial advances in this area. This will provide more efficient and cost effective implementations of tertiary-storage-backed database systems from an ECS perspective, regardless of whether this is the only storage mechanism supported by the architecture.

- **Incrementally define Earth Science type library**

  Categorization: ECS, PDR, EVO (S/L)

Substantial extension to S2K, NetCDF and HDF metadata is required to be useful in EOSDIS. Development of the earth science type library is a major effort, and one that will require periodic refinement to ensure adequate support for science applications over the life of ECS. The current development effort includes definition of a core metadata model and design of a rich Earth Science Data Type (ESDT) taxonomy with associated services/methods. The ECS team should review this design in some detail at PDR, and should continue to evolve the types and methods as time goes forth.

- **Incrementally define SQL-* schema and data dictionary**

  Categorization: ECS, PDR, EVO (S/L), R&D

A data dictionary of common terms and synonyms to be used by EOSDIS scientists, and textual descriptions of data elements need to be defined. Moreover, an SQL-* schema needs to be defined in an iterative and evolutionary manner to represent EOSDIS data and to support the middleware. Techniques and tools to semi-automatically build and integrate these schema and the data dictionary are currently not well understood, even by the research community.

Again, this is being done within ECS in the data modeling (core metadata definition), Data Server design (schema definitions and extensions), and Data Management design (data dictionary) efforts. These designs should be reviewed at PDR. Additionally, the design needs to demonstrate the ability to adapt to schema and data dictionary changes over time—as new products are added, new attributes used to describe existing products, and new or modified terminology added to the data dictionary. Longer term R&D efforts should include development of tools for schema analysis, integration, and federation. This will support management of the federated/integrated schema hierarchy in a dynamic environment. Tools such as these are one of the targets for prototyping.

- **Client-middleware & middleware-server protocols (Gateways)**

  Categorization: ECS, PDR, EVO (S/L)

Such protocols will allow the inter-communication among Client applications, Middleware, and Database servers. SQL-* will be used as the intermediate language. The team discuss the interoperability and portability of two protocols: API and FAP in Section 3.5.1. Since SQL-* is proposed as the query language, a combination of APIs and gateways is needed to achieve vendor independence (portability and interoperability).

The team contrasts the two protocols with other client-server protocols, including Z39.50, CORBA, and HTTP (see Section 3.5.2). In summary, the protocols are very important to the success of building a large distributed information system that includes both legacy and new data servers. Current standards and gateway technology to facilitate interoperability among a number of different servers are still immature. Therefore, a close look at the protocol selection and standard is strongly recommended.

The current ECS architecture provides "gateway" functionality at the Data Server interface. The Data Server exports a list of protocols and services that it will respond to, and clients attach to those services using the appropriate protocols. The Data Server may convert the accepted protocol in any way it desires internally. This approach supports heterogeneous providers by allowing them to "advertise" services in a native protocol, rather than putting the burden on them to convert to a "standard" protocol. The ECS team's requirement to develop components for GCDIS reuse and to prepare for a heterogeneous environment has led them to this design point. The details of this design will be discussed at PDR. Extended type queries in the flavor of SQL-* or SQL3 will be an important part of this protocol suite, but they will most likely not be the only query and delivery protocol. The rapid acceptance and continued development of HTTP is likely to continue, and it is reasonable to believe that "providers" in the longer term will want to provide services via this and other protocols. Nevertheless, the ECS team needs to consider evolution of the system to a predominantly SQL-*/SQL3 based infrastructure, and show how the system can adapt to such a scenario.

- **Standards evaluation (SAIF, OGIS, OMG, SQL/MM)**

  Categorization: PDR, EVO (S)

ECS should make a major effort to influence these standard activities so they become even more useful over time. The ECS team needs to show at PDR its involvement in these and similar standards efforts, and demonstrate infusion of appropriate components into the core metadata and Data Server designs. Because standardization is a longer term prospect in a number of these cases, the ECS design needs to be able to react to changes in the status of these standards—e.g., how will widespread adoption of OGIS affect the core metadata model and Data Server designs.

- **Don’t use CORBA**

  Categorization: ECS, PDR, EVO (S)

CORBA (Common Object Request Broker Architecture) is a client-server request-reply protocol defined by the Object Management Group (OMG). It is a standard way for a client to invoke a server. The team enumerates several reasons why CORBA should not be used as currently defined, see Section 3.5.2. ECS needs to monitor this standard closely.

The approach to CORBA adoption is outlined in the ECS System Design Specification, and will be addressed again at PDR. The current approach employs DCE in the release A/B timeframe, with migration to CORBA in the release C/D timeframe as the technology matures. The "Don't use CORBA" argument is addressed at the fundamental search mechanism supported in the system. The UCB team suggests that a CORBA foundation will lock the ECS design out of commercial advances in cross-site, distributed search, and force development of similar functionality on a CORBA foundation. ECS needs to address this concern by showing how the architecture can evolve to support the commercial distributed search capabilities that will evolve on a SQL-*/SQL3 foundation.

- **Data lineage support**

  Categorization: ECS, EVO (S/L)

Heritage of the data need to be captured and manipulated using a DBMS.

The current ECS architecture supports capturing of data lineage information. This will most likely be done in a DBMS environment, as suggested by the UCB team. Within ECS, lineage will be explicitly tracked through production "logging" to the appropriate information repository. The UCB team points out the benefits of database technology in supporting automatic lineage capture through mechanisms such as DBMS triggers and audit trails. Use of such mechanisms should be addressed in discussions of system evolvability, ensuring that the overall system can take advantage of these capabilities within various Data Server implementation strategies.

4.1.2 Network Management

- **Use DCE**

  Categorization: ECS, PDR

DCE (Distributed Computing Environment) is network operating system software that supports a client/server model across platforms. The software is currently supported by many vendors.

The use of DCE is recommended for lower level network services and network management. Given the SQL-* approach, this will require SQL-* middleware vendors to build distributed query processing and schema integration/analysis capabilities on a DCE-based infrastructure to provide network security and inter-site/process communication for heterogeneous platforms. Presumably, DCE/RPC is not proposed as the query and response delivery mechanism, as it has a similar request/reply model to that of CORBA. This will require a DCE-based approach to the SQL-* middleware layer to implement the approach proposed by the team. The ECS team has adopted DCE as both the network delivery (request/reply) and network management layers for release A/B, and should be prepared to address the implementation approach at PDR.

- **Adopt System Network Management (SNMP)**

  Categorization: ECS, PDR, EVO (S/L)

SNMP is a set of tools and standards that allows the systems to scale to thousands of nodes and still be manageable. Netview product suite is identified as one of the tools.

ECS has standardized on SNMP. The design details should be reviewed at PDR, as should plans for taking advantage of evolution in network management technology.

Other Architectural Issues

- **Should do “just-in-time” hardware acquisition (6 months)**

  Categorization: PDR

The report presents the team's assessment of technology trends in CPU, disk, tape, and network technology. The team bases its cost models on "just in time" hardware acquisitions (6 months prior to operational use) to reduce hardware costs.

Current ECS acquisition is phased as suggested by the UCB team. However, much of the ECS hardware architecture is based on "high end" components (FDDI LANs, HiPPI, shared disk, etc.), which may have longer procurement cycles than commodity technologies. The ECS team acquisition strategy needs to be reviewed at PDR. This should cover the current acquisition strategy and basic assumptions, including the phased capacity planning (i.e., production demand basis), and procurement constraints.

- **Use automated techniques to reduce user support**

  Categorization: ECS, PDR, EVO (S)

The team proposes to move from a “human-intensive” support system to an “electronic-intensive” support system using on-line help, Mosaic and the World Wide Web (WWW).

The current ECS plan is to utilize electronic support mechanisms to the extent possible. This includes on-line help facilities accessed through standard interfaces such as WWW browsers. These plans should be reviewed at PDR. Evolvability to take advantage of advances in these areas should also be addressed.

- **Share DBAs among superDAAC and peerDAACs**

  Categorization: PRG

The UCB team recommends a review of staff assigned to DBA-like activities, suggesting that a single individual can support a large number (on the order of 25) of smaller facilities. It is not clear how this recommendation relates to the current ECS approach, which has DBAs concentrated at the DAACs. Programmatic review of this distribution is suggested.

- **Email support, electronic documentation ....**

  Categorization: ECS

Electronic documentation, help and email support are all planned within the current ECS approach. The ECS team should continue to look for ways to efficiently support the user base, using electronic means to leverage human effort.

- **Technology and user scenarios assessment**

  Categorization: ECS, PDR

Several valuable user scenarios (*original document* Section 2 and Appendix 3) and technology forecasts (*original document* Appendix 2) are described in detail in the report. The database schema design and SQL-* queries presented in support of the user scenarios are described in *original document* Appendix 4. The team also presents a risk assessment of reliance on database technology in Section 1.10. We highly recommend that the ECS team take a close look at these studies. Implications of these scenarios for the current architecture should be addressed at PDR.

- **Cost assessment (advanced technologies, rearchitect DAACs, ...)**

  Categorization: PRG

Section 5 presents a 6-year budget of the cost to procure and operate the main components of the proposed architecture based on the prediction of advanced technologies in software/hardware, and on the redistribution of functionality among the "2 + N" DAAC model.

We recommend a programmatic review of some of the approaches suggested by the UCB team. In particular, consideration should be given to streamlining the procurement process and hence supporting "just in time" acquisition. Additionally, the program may want to look at ways to distribute functionality and move into more commodity technologies as suggested by the peerDAAC concept. We realize that this could have significant programmatic considerations, and may not be achievable in the short term.

- **Influence small to mid-size COTS vendors & minimize in-house development**

  Categorization: PRG

EOSDIS should focus on working with COTS software rather than develop large proprietary systems. The team recommends that ECS should work with at least two SQL-*/HSM database vendors and should purchase COTS system management tools based on the SNMP protocol.

This appears to be an effective approach to developing some of the larger system components that are well aligned with commercial product interests. The program should find ways to accelerate development through outsourcing where program technology interests are suitably aligned with commercial interests. This is particularly true in the marriage of database and HSM technologies, and in the development of distributed query capabilities.

- **ESN between superDAACs and Internet for peerDAACs (bandwidth & cost ....)**

  Categorization: PRG

The proposed architecture relies on the rapid improvement in network throughput and reliability. This recommendation is consistent with the current approach of providing ESN interconnection between DAACs, some high speed links to high volume QA facilities (SCFs), and Internet connectivity to the community at large. The ramifications of providing high speed ESN connectivity to a smaller number of DAACs are a programmatic issue.

4.2 Summary and Analysis of the GMU Independent Architecture Study

The George Mason University (GMU) team was composed of an interdisciplinary group of earth scientists and computer scientists. The team identifies a number of sources of stated earth scientist's needs, including some of the key sources developed by the ECS team in its architectural redesign effort. The GMU team adds their own observations, and develops a broad architectural framework capable of supporting these needs. Like the current ECS approach, much of the focus of the GMU team was aimed at developing a flexible architecture for ECS that can accommodate a growing level of demand from both science and non-science users.

The fundamental solution recommended in this report is to create facilities referred to as Domain Application Data/Distribution Centers (DADCs) which can service the unique needs of specialized user communities. While the establishment and funding of such facilities is a programmatic issue, the GMU team addresses the ability to support such a concept from an architectural perspective. This is quite similar to the ECS team's approach at developing a "policy neutral" architecture that allows data provider policies to be established independently of the core architecture development.

As a result, the GMU team has arrived at an architectural framework that is conceptually similar to the current ECS design, with the built-in flexibility to meet the requirements of ECS under the current programmatic framework, but which also will be able to evolve to support the inclusion of a broader group of researchers and users whose needs are not specifically addressed by the current scope of EOSDIS. Ultimately, the inclusion of these communities will also return value back to the science users of EOSDIS by providing them with more ready access to data products that are generated outside of the normal "production" system.

The earth scientists on the GMU team explored several scenarios in each of three earth science research areas: terrestrial, oceanographic, and meteorological. Each of these scenarios describes a variety of aspects of user interaction with the system that have important implications for ECS architecture designers. Not only do these scenarios describe various types of query sequences that users will make individually, they also describe various forms of collaboration and data sharing between users that must be supported by the architecture. Throughout these scenarios, one theme seems to dominate: most serious earth-science research will require a combination of data from sources both inside and outside of the supported EOSDIS infrastructure. It is therefore vital to these scientists that the ECS architecture provide them with an effective means to access all relevant data, regardless of the source. Another item worth noting is an anecdotal description in Part 2, Chapter 5 of a case in which an algorithmic flaw created erroneous data that propagated errors into many high-level data products. The discussion describes a number of issues that must be considered in regard to identifying, tracking down, and rectifying such inevitable errors.

The computer scientists on the GMU team developed a federated client-server architecture that is intended to address the key science drivers identified by the earth scientists. An important aspect of the client-server architecture is that it is composed of standardized "ECS nodes" which may be individually configured to meet varying levels of processing and archival requirements.

One of the more unique aspects of the GMU team's information architecture is the emphasis on incorporating and organizing knowledge in the system in addition to data. This provides for the organization of "info-marts" for value-added, knowledge-rich specialized products. Issues of knowledge-based information access are addressed through the notion of a Global Thesaurus. Tying it all together is their EOSFed concept which describes the structure and functions needed to allow autonomous and heterogeneous systems to become EOSDIS members. These concepts directly parallel the concepts of value-added providers, domain specific data servers, and smart data dictionary present in the ECS architecture.

Other important aspects of the architecture development were the modeling and analysis techniques used by the GMU team. Custom-modified COTS domain analysis tools were applied to characterize a family of systems that are independent of a specific mission, instrument suite, or hardware configuration. Using this domain model in combination with a functional description of the architecture, a performance modeling methodology was used to illustrate how a proposed architecture can be analyzed. Using these analysis techniques, it is possible to determine the impact that various alternative system configurations might have on response time. Another important form of analysis provided in this report is a set of event-sequence diagrams of various user scenarios, showing how the proposed system responds to user requests. GMU recommends that the modeling approaches and expertise of the GMU team could be used to augment ECS modeling efforts.

To complement the architecture development, studies are provided on ATM networks, data storage systems, and high performance database management systems. These studies outline the current state of the art, and make specific recommendations for the ECS architecture. These recommendations are, in large part, consistent with the ECS team's stated technology adoption plans as put forth at the System Design Review (SDR).

Finally, the report also addresses issues of management of large science projects and how to effectively integrate EOSDIS into the broader context of projects such as GCDIS. The report stresses the need to make data and services available to the widest possible user population in order to achieve maximal success for the overall program. This was a common theme from the teams, and suggests that the ECS team should be quite aggressive in defining its value-added provider (VAP) interfaces.

Below are some of the key recommendations for the ECS program based on the GMU report:

4.2.1 Information Management

- **Re-architect DAAC roles and add DADCs to provide specialized support consumer-oriented services:**

  Categorization: PRG

A primary element of the GMU approach is the notion that the architecture must be able to support Domain Application Data/Distribution Centers (DADCs). The role of the DADC is to provide services and high-level EOS data products to specialized communities of users. The DADCs will serve both to offload the DAACs from intensive pull demands from thousands of non-science users as well as to improve the access to non-EOS data for the science users. (Part 5, Ch. 1).

This concept is consistent with the ECS architecture concept of "Value added providers." The ECS architecture envisions SCFs and other data centers as being able to fulfill this role by taking primary feeds from the DAACs, repackaging data and services, then delivering those to its own community through a variety of interfaces. The movement of production responsibility for the "standard products" between DAACs, SCFs, and other providers, is a programmatic/policy issue.

- **Use of Global Thesaurus concept:**

  Categorization: ECS, R&D, EVO(S/L)

The GMU architecture introduces the concept of a Global Thesaurus whose purpose is to provide a mapping from a user's term into both broader and narrower terms, thereby increasing the likelihood of obtaining the desired information from a repository of information holdings. Unlike a standard thesaurus, the Global Thesaurus makes use of active rules to aid in the translation of terms in different domains. (Part 3, Ch. 4, pg. 7)

This concept is partially accommodated by the ECS architecture's Data Dictionary component. However, the GMU team proposes an innovative approach to thesaurus configuration and management at both the local and global levels of the system. Their recommendations should be investigated further as part of the ECS design. The feasibility of establishing a truly global thesaurus remains to be established. The successful implementation of this concept hinges on a series of technical, behavioral and managerial issues.

- **Enhancement of data browsing and query processing:**

  Categorization: ECS, EVO (S/L)

The primary role of the Global Thesaurus is to improve data browsing and query processing functions. Queries made by the user will often not match items in the database, simply because the terms used in the query are not the same as those associated with the data. The Global Thesaurus will make this access more effective. Some of the issues involved in providing an intelligent thesaurus require complex semantic mappings. Such issues will be addressed in an evolutionary fashion within ECS.

- **Scalability of Global Thesaurus to a large distributed system:**

  Categorization: R&D

While GMU has initiated work on a Global Thesaurus, further research is needed to demonstrate that this concept can be scaled to a large distributed system. A number of the research issues that must be addressed are similar to those of large scale, distributed information systems, with added complexities in knowledge representation, concept definition and refinement, etc.

- **Creation and maintenance of the Global Thesaurus:**

  Categorization: R&D, PRG

Further research is also needed to show how a Global Thesaurus might be created and maintained by a large scientific community. The more people that get involved in this process, the more difficult it becomes. From a research perspective, there are three fundamental problems. Tools are needed to semi-automatically capture domain concepts and the knowledge that links them together. Also, tools are needed to maintain consistency of the knowledge. These are difficult problems even without distributing the knowledge base, so implementing a global solution in a distributed environment will be particularly challenging. Additionally, there will be programmatic issues involved in resolving semantic terms and mappings proposed by different segments of the ECS user community.

- **Use a federated architecture to allow integration of heterogeneous components (multi-database support for the EOSFed concept):**

  Categorization: ECS, EVO (S), R&D

The GMU approach advocates a federation of heterogeneous and autonomous information systems. Therefore, they have concentrated on issues related to supporting varying degrees of coupling to the federation, management of the federation, and provisions of mediation services that allow translation of information between a local information system and the rest of the federation. (Part 3, Ch. 4, pg. 31)

This recommendation is consistent with current ECS plans to support heterogeneous, autonomous providers. Federation approaches and issues raised by the GMU team are incorporated in the short and long term design plans of the ECS team.

- **Pre-query cost estimation using meta-data:**

  Categorization: ECS, EVO (S/L)

If the size of a response to any query exceeds a user-determined threshold, the GMU architecture is designed to provide a report to the user indicating the time that would be required to deliver this information as well as options for different delivery alternatives with their associated costs. (Part 3, Ch. 1, pg. 5)

This recommendation is consistent with current ECS plans to support provider service descriptions (including costs), as well as service-based accounting and mediation services. The

GMU team's suggestions should be further incorporated into the ECS team's design and implementation efforts.

- **Provide interoperability with data sources outside of EOS:**

  Categorization: ECS, EVO (S/L), PRG

GMU scenarios emphasize the need for scientists to be able to easily access data sources outside EOS. The DADC concept recommended by GMU provides a means to support this outside access.

A significant degree of external system interoperability is incorporated in the ECS architecture, primarily as a means of accessing ancillary and correlative data for purposes of product generation. ECS is required to be interoperable with Ancillary Data Centers involved in production processing. The GMU team recommends a broader definition of interoperability, one that would enable users to make transparent queries across multiple data and information systems (ECS plus other systems). In addition to requiring technical work to support this in an evolvable fashion, this issue requires some programmatic direction from ESDIS.

- **Generic ECS node architecture that can be specialized to support different user requirements:**

  Categorization: ECS, EVO (S)

The building block of the GMU architecture is called an ECS node, which is a collection of servers of different types. ECS nodes can be configured to serve as DAACs, DADCs, ADCs, or even SCFs. (Part 3, Ch. 2, pg. 6)

The GMU team specifically decomposes the ECS node into distinct server "types," which it appears to make available through site interfaces. The ECS architecture supports a similar implementation, however it strives to unify the interface into those servers through a common gateway interface called the Data Server. This provides the "federation" mechanism alluded to by the GMU team, and allows individual sites to decide which servers and services it wishes to make available.

- **Archival of data in a variety of user-oriented forms:**

  Categorization: EVO (S)

GMU proposes the Info Mart and Data Warehouse concepts as a means of extracting and organizing data according to a common schema so that specific user communities will be able to access the data for decision making, and to encourage the production of domain-specific value-added products. (Part 3, Ch. 4, Pg. 12)

In fact, this concept is also consistent with ECS architecture and evolvability approaches. Value added providers will receive primary feeds from the ECS DAACs, and will then organize the data according to the needs of their own communities.

- **Layering of the architecture to separate push and pull requirements:**

  Categorization: EVO (S), PRG

One of the key points behind the use of DADC facilities is that the DADCs can service the specialized pull requirements of specific user communities, thereby allowing the DAACs to focus on their role of handling the push requirements of data processing and archiving.

Again, such separation is achieved in the ECS architecture through the Value Added Provider (VAP) network. The development and evolution of VAPs over time will be driven by individual user community data needs. The provision of Government funds for the creation of VAPs is a programmatic issue.

- **Methodology for performance analysis of large distributed client-server architectures (do performance modeling scientifically):**

  Categorization: ECS, EVO (S)

GMU has applied a mixed queuing network modeling approach for analyzing performance of prospective architectures. This technique permits scientific studies of different architectural alternatives, and provides quantitative estimates of system response time and sensitivity to changes in workload. (Part 3, Ch 5)

The ECS team should augment its performance modeling activities to include the sorts of studies advocated and demonstrated by the GMU team, especially for the "pull" side of the system.

- **Query formulation and optimization that uses metadata and thesaurus information:**

  Categorization: ECS, EVO (L), R&D

Provide cooperative query formulation services in which metadata and knowledge are used to assist users in formulating and reformulating requests. Allow for both generalization and specialization in both the temporal and spatial dimensions. (Part 3, Ch. 4, pg 19)

- **Construction and use of query optimization rules (for extended types)**
- **Cooperative query formulation**
- **Automatic request formulation**

These are longer term R&D issues that the ECS team needs to pursue. The ECS team should show how developments in areas such as these fit into the overall architectural strategy, and allow long term system evolution.

- **Apply domain modeling tools to assure proper system interaction:**

  Categorization: ECS, EVO (S/L)

Support systematic evolutionary development of the architecture by developing a domain model that describes the problem-oriented aspects of the architecture that remain relatively constant throughout the life of the system. (Part 3, Ch. 3, pg. 2)

In fact, this has been done through a series of "logical object models" that formed the basis for the ECS System Design Review. The GMU team recommendation should act as a reminder to the ECS team to recheck these domain models periodically to ensure their constancy, and to guide detailed design decisions.

- **Location-independent names are mapped into location dependent names to allow ECS objects to be easily moved:**

  Categorization: ECS, EVO (S)

The GMU architecture primarily operates with location-independent names, called logical names and mapped names. Logical names provide terms that are meaningful to users, while mapped names are meaningful internally to the ECS Services layer. Special naming and catalog services are used (within the distributed object management layer) to convert mapped names into physical names that correspond to distinct locations. Nowhere within the ECS Services layer are any physical names used. (Part 3, Ch. 1, pg. 8)

This recommendation is generally consistent with the ECS architecture. Location independence is obtained within ECS through both logical service names and service attributes, which can be employed in service "searches". This approach allows for the evolution of available services and the migration of services in response to system loads or user needs. The GMU team proposes the creation of location-independent names for both logical and mapped services. This is an alternative approach to achieving location independence which is not excluded by the current ECS architecture.

- **Cache user information at client nodes:**

  Categorization: ECS, EVO (S)

A user profile manager keeps track of user histories so that users can continue sessions from where they stopped previously, and so that a user's preferred areas of interest can be monitored. (Part 3, Ch. 2, pg. 3)

In the ECS architecture, these features are implemented through stateful sessions between the client and the DataServer/distributed data management system, and through desktop installation of customized or specific services. The user profile parameters that will be captured during a user session have not been fully defined. Furthermore, the degree of session-to-session continuity provided by the system remains to be defined.

- **Cache Global Thesaurus data in a Local Thesaurus at client nodes:**

  Categorization: EVO (L)

Portions of the global thesaurus information are cached locally at each user's node. An "optimistic cache coherence protocol" is used to update a local thesaurus if its information should happen to become out of date. (Part 3, Ch. 2, pg. 4)

These concepts should be integrated into the ECS distributed data dictionary. Because "concept" caching is a bit more difficult than simple term caching, the ECS architecture will need to be flexible enough to evolve to this scenario.

- **Support rules as first-class objects:**

  Categorization: R&D

The system must be able to represent rules in such a way that they can be associated with different objects at different times. Therefore, rules must not be represented as attributes of other objects. (Part 3, Ch. 4, pg. 37)

In particular, the ECS team should consider rule-based extensions to their current data dictionary. This will allow for the development of global thesaurus capabilities such as described by the GMU team. The use of condition-action rules in production management has been proposed by the ECS team. The GMU team advocates a more fundamental inclusion of these concepts in the core architecture.

- **Reconfigurable node architecture:**

  Categorization: EVO (S/L)

Design the ECS node architecture such that it may be configured in a variety of ways from a standard set of component objects.

This is precisely the reasoning behind the object-based Data Server design of the current ECS architecture. It allows providers to develop as little or as much functionality as they wish, employing "inherited" code where possible in the development of their services.

- Generated from domain model specification:

  Categorization: R&D

GMU proposes the use of their domain modeling tool which combines a standard CASE tool with a knowledge based tool to elicit from the user target system requirements. Ultimately, this could be used to automatically configure an ECS node from a set of standard component modules. (Part 3, Ch. 3, pg. 29)

While some aspects encompassed by this recommendation are quite "researchy", others are nonetheless more pragmatic. For example, it is conceivable that an Earth Science Data Type library could be developed in such a fashion as to support CASE-based configuration as the GMU team suggests.

- **A variety of information management services are recommended:**

  Categorization: ECS, EVO (S/L), R&D

The ECS node architecture is composed of over 15 distinct servers that address a variety of information processing tasks. The functions of each of these servers are described. Some of these servers are well within current technology, while others may require further research before they can be realized in practice. (Part 3, Ch. 2)

The ECS information management services are provided through a common gateway that then maps to the implementation "servers", as described by the GMU team. The sophistication of information management services provided by ECS will progressively expand through the four major releases planned through 2002.

4.2.2 Network Management

- **Use Virtual Client Protocol for reduced network traffic:**

  Categorization: ECS, PDR

Reduce network traffic by designing a protocol which allows a secondary server to return its response directly to the primary client, rather than relaying the data through an intermediate requestor. (Part 3, Ch. 2, pg. 24)

This is envisioned for ECS, as outlined in the ECS end-to-end scenarios presented at SDR. The implementation details will be presented at PDR.

- **Make use of ATM switches:**

  Categorization: ECS, EVO (S)

The overall network design should have the wide area network supported with ATM technology. Various tradeoffs must be considered to determine whether ECS should own its network or purchase use from a public carrier. (Part 6, Ch. 1, pg. 12)

ATM technology is being investigated by both ESDIS and the ECS team as the likely technology for DAAC internetworking. Considerations for more widespread access, including ATM-to-the-desktop, are evolvability concerns as the Internet undergoes infrastructure upgrade. They may uncover programmatic and cost issues as well.

- **Be CORBA compliant (no rationale given): (Part 6, Ch. 3, Section 3.8)**

  Categorization: ECS, EVO (S/L)

The ECS service infrastructure is CORBA-based, with shorter term implementations based on DCE and longer term migration to fully compliant CORBA infrastructures.

- **Manage future consolidation and reallocation of DAAC resources:**

  Categorization: EVO (L), PRG

Network infrastructure should be kept sufficiently flexible so as to accommodate future changes in DAAC responsibilities and processing loads.

In order to support longer term reallocation of services amongst DAACs, the network infrastructure should be developed in a modular fashion. This would allow reallocation of resources as necessary to accommodate changes in DAAC responsibilities. Programmatic support for this concept needs to be developed consistent with the technical approaches.

- **Be able to exploit idle computing resources on the network:**

  Categorization: EVO (L), R&D

Non-ECS sites can register idle CPU cycles with the ECS Product Scheduling Servers so that non-ECS sites can be used to process or reprocess standard data products (Part 3, Ch. 2, pg. 28)

In fact, this is addressed in the ECS Conceptual Architecture, where NSF supercomputer center resources are identified as candidates for addressing peak processing needs in response to phenomenological events. The current ECS concept supports the "advertisement" of such resources into ECS. The implementation of these services needs to be addressed.

- **Separation of data and control paths:**

  Categorization: ECS

Use an FDDI LAN for control messages and a high speed HIPPI channel for data transfers. (Part 6, Ch. 2, Section 2.2.4)

The importance of this design concept is recognized by the ECS team. E.g., most of the COTS storage and processing products being evaluated by the ECS team employ this design concept.

- **Network-attached storage:**

  Categorization: ECS

Make use of new technology which allows storage device controllers to be connected directly to a network so that data can flow directly from the storage device to the client application without going through the server computer's memory. (Part 6, Ch. 2, Section 2.2.3)

This recommendation is consistent with current ECS implementation plans.

4.2.3 Other Architectural Issues

- DADC Concept (Part 5, Ch. 1)
  - provide multiple access points to EOSDIS
  - integrate EOSDIS with GCDIS under framework of UserDIS
  - provide interoperability between different domains
  - provide higher level of information management
  - support higher level data products
  - provide domain-based support services
  - provide customized user interfaces
  - establish domain communities and domain networks
  - manage federations of user communities
  - develop common community standards

– provide specialized sales and accounting procedures

– organize domain-specific knowledge

Categorization: ECS, EVO (S/L), PRG

The DADC concept in general, as well as the specific items listed above, are consistent with the current ECS architecture. Multiple access points are provided through heterogeneous services and interfaces, including direct query interfaces, WWW-based hypertext interfaces, and text-based interfaces. EOSDIS components are being developed with longer term GCDIS and UserDIS in mind. Domain interoperability and higher level information management are provided through the development of DIM (distributed information management) services. Higher level data products are supported through SCFs as data providers and additional, value added providers. Domain-based support services are supported through the existing DAACs, and extended as necessary by SCFs and value-added providers. Customized user interfaces and specialized accounting procedures are also part the this concept. Finally, the concept of a federation of domain communities with domain-specific knowledge, all tied together by common standards is consistent with the ECS concept implemented through Data Server, LIM, and DIM services. The creation of an external DADC-type interface to the ECS is a programmatic issue. Funding for the implementation of DADC organizations is also a programmatic issue.

- **Consolidation of DAACs:**

Categorization: PRG

Many alternatives for the evolution of DAACs exist. GMU suggests that the evolution to four main DAACs with the others becoming SCFs is likely. (Part 5, Ch. 1, pg 10)

The existing DAACs have been selected based on historical support to domain specific users. Changes in the responsibilities of these facilities are a programmatic issue, though the ECS team has identified this as an evolutionary test.

- **Decouple non-earth scientists from direct DAAC access:**

Categorization: EVO (L), PRG

In order to relieve the DAACs from the potentially tremendous PULL demand from non-science users, it is recommended that intermediate facilities such as DADCs be used to service these users.

The ECS team's Value added provider (VAP) concept is intended to address this concern. Funding for the implementation of VAPs is a programmatic issue.

- **Coordination of GCDIS with EOSDIS:**

Categorization: EVO (S/L), PRG

GCDIS offers the opportunity for NASA to significantly enhance the value of EOSDIS to the user community. Software developed for EOSDIS may also reduce costs for GCDIS prototype development. (Part 5, Ch. 2)

This was one of the GCDIS "drivers" for EOSDIS, and has been considered in the ECS architecture. This will require continued coordination between the ECS team and the larger GCDIS community to ensure adequate functionality for GCDIS.

- **Storage system assessment (Part 6, Ch. 2) explore high performance storage systems (HPSS):**

  Categorization: ECS, EVO (S), R&D

Consider use of a scalable distributed file and storage management system that can handle large volumes of data and handle high data transfer rates (Part 6, Section 2. 3.1)

The ECS team has a number of evaluation and prototyping efforts underway and planned for the evaluation of large mass storage systems. The team has also planned on mass storage technology developments that will require archive technology phasing over time.

- **Parallel database management and information retrieval systems prototype using Oracle and UniSQL: (Part 6, Ch. 3, Section 3.8)**

  Categorization: EVO (S), R&D

Database management systems will evolve in a number of directions, including increased parallelism, and extensions into object oriented implementations. The ECS team is planning for such advances, evaluating available databases as they are available, and integrating them into the architecture at the appropriate time.

- **Thoughts regarding management of "large science" projects:**

  Categorization: PRG

The failures of previous large science projects are examined, and new principles for managing large science projects are considered. (*original document* Appendix B)

The GMU team's experiences in this area are quite useful. The observations and recommendations of the GMU team have been communicated to NASA managers within the GSFC ESDIS Project Office and the Mission to Planet Earth Program Office at NASA Headquarters.

- **Analysis of the scalability of the push-pull model:**

  Categorization: ECS, PDR

The properties of PUSH and PULL are examined and compared. (*original document* Appendix B, pg. 14)

The GMU team's analyses should be addressed along with ECS team modeling at PDR.

- **Process level 4 data at DADCs, levels 0-2 at DAACs, and level 3 at both:**

  Categorization: PRG

This will keep most of the push functions at the DAACs while domain-oriented knowledge and data sets reside at DADCs. (Part 5, Ch. 1, pg 5)

The ECS architecture would support such a configuration, although current programmatic assignment places these production responsibilities at the DAACs (levels 0–3), the Data Assimilation Office (for level 4 products defined within the current scope) and the Science Computing Facilities (for special products at levels 1–4). The reallocation of production processing responsibilities to different organizational entities is a programmatic issue.

- **User Scenarios:**

  Categorization: ECS, PDR

User scenarios are provided for terrestrial, oceanographic, and meteorological studies. Each scenario has quantitative estimates of the data volumes that will be involved. Also, user access is examined and estimates of intra-arrival times for queries are calculated. (Part 2)

The ECS team should address GMU team scenarios at PDR.

- **Technology Assessment:**

  Categorization: ECS, PDR

Technology assessments are provided for ATM, Data Storage, and High Performance DBMS technologies. (Part 6)

The GMU team's technology assessments should be integrated into ECS technology assessments, and addressed at PDR.

- **Lessons from the Human Genome Project:**

  Categorization: ECS, PDR

A summary of lessons learned from the Human Genome project is provided. This is of extreme relevance to EOSDIS, because it too is a large-scale data intensive scientific endeavor. While the overall volume of data is much less in the Genome project than it is for EOSDIS, issues of data access and user collaboration are still of prime importance. (*original document* Appendix A)

The ECS team should integrate lessons learned from the Human Genome Project, and should find additional projects of relevance that share the characteristics of a large-scale scientific information system.

4.3 Summary and Analysis of the UND Independent Architecture Study

The UND team's focus for the ECS Independent Architecture Studies was on the infrastructure, both programmatic and architectural, needed to support their estimate of millions of potential ECS users in the general public. The primary component of their architecture was the Public Access Resource Center or PARC, which is a value-added service that can be layered onto the

ECS architecture but is currently not within the scope of the ECS program. The UND team recommended that EOSDIS plan to support millions of users based upon current levels of data utilization at existing climate and agricultural research centers and the rapid growth of Internet Mosaic users during the past few years. The primary guiding principle for their architecture was to allow user demand to drive system evolution in a manner similar to natural selection, that is where the PARCs that are successful will survive and others will die out. They recommended that COTS be used wherever possible to support the broadest community of users at the lowest cost, and that direct broadcast of data be a long term commitment of ECS and NASA in order to ensure timely access to data at the lowest cost. They presented a prototype PARC for agriculture called an AgPARC, that was based on a system currently in use at the Regional Weather Information Center (RWIC) at UND and several scenarios primarily in the agricultural user community which would benefit from EOS data. They recommended further that EOSDIS begin an educational effort aimed at informing the general public of the potential benefits of EOS data and of the EOSDIS program and provided a sample Mosaic home page to accomplish this for agriculture users. Finally, they provided data on network performance in real testbeds indicating that there are several variables that must be addressed to determine end-to-end application performance, and they used the BONeS network simulation tool to compare various topologies for interconnecting the DAACs including the Version 0, hypercube, mesh, and fully connected topologies.

Note: In the following discussion, we attempt to tie recommendations to potential implementation of the concept of PARCs as a primary interface to the ECS for non-science users and as an intermediary component of the overall EOSDIS architecture situated between ECS and the general public. The basic premise and approach of PARCs as funded entities within EOSDIS is a significant programmatic issue that we do not address further.

4.3.1 Information Management

• **Layer on top of current ECS architecture:** The UND team study focuses on the development of PARCs that will be layered on top of the current ECS architecture that supports archiving at the DAACs. This layering is viewed as both a short and long term evolution and depends on the applications domain needs of the non-science users.

Categorization: EVO (S/L)

This concept is consistent with the ECS architecture concept of "Value added providers" (VAPs). This concept supports VAPs with both a subscription-based and a query-based interface into ECS holdings, and allows VAPs to generate derivative products and distribute them freely to the community. Initially, many of the NASA-funded Science Computing Facilities (SCFs) are expected to function as VAPs. As time goes forth, a number of additional discipline-oriented secondary providers are likely to arise, and the ECS architecture needs to demonstrate scalability of the secondary distribution mechanism that will support this growth.

• **PARCs use client server model:** The obvious choice for development of a PARC in both the near and long term is client-server (p.36)

Categorization: EVO (S/L)

The secondary provider interfaces need to be developed to support client/server interaction between the secondary providers and the core system.

- **PARC interoperability: Has implications for compatibility with ECS.**

SQL-2, OSI: Recommended for development of a PARC today (p. 36)

DCE: Recommended for use today (p. 36)

SQL-3, COTS (Sybase Enterprise Manager): Recommended for development of a PARC within 2-5 years. (p.36)

Object-oriented products (CORBA, objectbase systems): Recommended for development in the five year time frame (p. 37)

Categorization: ECS, PDR, EVO (S/L), R&D

In an overall sense, none of these items directly address ECS, as they are guidelines for technology implementations of PARCs, which currently fall outside of the scope of ECS. Hence, PARCs will be free to choose whatever implementation strategy they desire. This is consistent with the heterogeneous provider foundation of ECS. In order for PARCs to interoperate with one another effectively, they may desire to select implementation technologies that are well aligned with ECS implementation plans. These recommendations of the UND team are consistent with ECS technology implementation plans.

- **PARC data access**

Standard indexed and access methods: Those available from COTS vendors for use today in the short term (p. 37)

Categorization: ECS, PDR, EVO (S)

These recommendations for PARC architectures are a bit more conservative than current ECS plans. Early releases of ECS will most likely employ type-extended database systems, in addition to currently available relational products. As these products mature and become more widely adopted, we would expect PARC implementations

Specialized utilities such as residual bit vector indexes: For longer term implementation for purposes of query acceleration; developed in conjunction with COTS products (see Section 3.4.2).

Categorization: EVO (L), R&D

The team recommends looking at new indexing techniques to speed query execution. Residual bit vectors are just one area ECS should explore. Others include spatial indexing techniques, feature-based indexing techniques, etc.

- **PARC-IMS search and access use Mosaic-like interfaces**: Hypertext point and click interfaces such as Mosaic are adequate for most users for the foreseeable future. Universal, compatible interfaces between different user communities (e.g., agriculture and education) are not needed - depend on COTS instead(p. 48)

Categorization: ECS, EVO (S/L)

The UND team advocates the use of COTS Web client technology for accessing ECS data. ECS currently plans on using WWW-based browsers like Mosaic, extended with service-specific interfaces to support, e.g., more complex search criteria, and a more dynamic interaction (stateful sessions). This is necessary to support some of the "higher end" ECS user requirements.

- **PARC data management software**

Sybase and Arc/Info: For implementation of a PARC today use the most capable COTS technologies available (p. 43).

Arc/Info and Object/Relational Product: In the short term use upgrades of Arc/Info and plan to move toward use of an object/relational product, but suspend choice until O/R products mature (p. 43).

Integrated GIS, DBMS, Objectbase mngt, visualization: For long term use develop integrated systems based on openness, capability, viability, compatibility with COTS (p. 44).

Categorization: EVO (S/L)

PARC data management software will most likely be based on current implementations of Arc/Info and Sybase. These products will improve over time, and will evolve to support extended data types and a more unified geospatial data model. ECS technology evolution is consistent with this recommendation, albeit at a more aggressive pace. In particular, the ECS data model will look to merge GIS and relational models through a geospatial data model. ECS is tracking efforts such as OGIS (Open Geodata Interoperability Specification) to plan for evolution to standards-based products.

- **PARC DASMA catalogue should use COTS data management products such as DCE for distributed catalogue management:** Catalogue managers accompanying COTS data management products will be adequate for PARC implementation in short term. DCE recommended.

Categorization: ECS, EVO (S)

The UND team recommends COTS-based catalogue management approaches. This is consistent with initial Advertising service implementations within ECS, which will employ Web-based browser access to relational implementations of advertisements through an HTTP gateway. The underlying mechanism by which distributed advertisements are accessed and searched will be described at PDR.

- **Proto AgPARC example design of PARC for EOSDIS:** An existing system is in use that implements some of the functionality needed for a PARC and can be useful as a model and prototype for further development, especially for agriculture.

Categorization: PRG

The ECS team should explore the prototype AgPARC as a driver for its value added provider (VAP) interface.

4.3.2 Network Management

- **PARC-DAAC connectivity**

T-1, T-3, internet: Recommended use of T1 and T3 for short term connection between PARCs and DAACs (p. 91).

ATM, B-ISDN: Rapid evolution of networks to use of ATM and B-ISDN in the next 2-3 years.

Based on NII: Long term recommendation is to support connectivity between PARCs and DAACs using the available National Information Infrastructure.

Categorization: EVO (S/L)

Current plans are to equip Science Computing Facilities with responsibilities for product quality assurance with network service to DAACs in proportion to their respective network needs. Other SCFs will rely on the Internet for connectivity with some upgrades supported by the ESDIS project. The ECS team's assumption is that newer networking technologies such as ATM will be used where possible and where cost effective in providing this bandwidth to SCFs. Additional PARCs will need to rely on NII upgrades to existing Internet technology under the current program structure.

- **PARC-PARC connectivity**

T-1, T-3: Same as PARC to DAAC above.

ATM, B-ISDN: Same as PARC to DAAC above.

Categorization: EVO (S/L)

Current network plans do not explicitly interconnect SCFs. These facilities, as well as other candidate PARCs, would communicate via Internet and the evolving National Information Infrastructure.

- **USER-PARC connectivity**

Dial-up phone lines (to 38800 bps): Implementation of user-PARC connectivity should be accomplished with existing phone lines today.

ATM, B-ISDN: Short term recommendation includes availability of ATM and B-ISDN for non-science users.

ISDN, B-ISDN: Reductions in cost and improvements in technology will make ISDN and B-ISDN a long term candidate (p. 92).

Categorization: EVO (S/L)

The UND team makes the point that many users of EOSDIS, and of value added provider services, will connect via "personal connection" networking technologies such as dialup lines and ISDN. Hence access to VAPs/PARCs will occur across a wide range of bandwidths, from dialup lines through ATM. It is recommended that the ECS team track the development of "personal connection" technologies to understand the capabilities of its user base over time.

- **ECS developers involved in ATM field trials:** To gain experience with actual network throughput, ECS should be involved in field trials.

Categorization: ECS, PDR

This is being done in conjunction with the ESDIS program office, and with a number of ATM networking testbeds that are in existence.

- **NFS outperforms custom RPC (should be used when available):** This recommendation was made in the context of supporting connectivity between an existing proto-AgPARC (at RWIC) and users. NFS outperforms in the following ways: portability across UNIX-based systems as well as MS-DOS based microcomputers; fails gracefully; controlled access to files.

Categorization: PDR

While NFS provides a more portable "interface" to network-shared data than does RPC, there are a number of problems with it in a WAN environment. Performance of NFS in a "slow network" environment, such as is often realized in cross-country Internet links, is poor. And although the file transactions may be handled gracefully in a volatile network environment, there are some issues with support for multi-user locking. Finally, access is controlled through the Unix rwx/ugo permissions, which may not be appropriate for large, complex data objects, where granularity of access may need to be managed at a "subfile" level. ACL-based services implemented through the RPC mechanism can avoid a number of these problems, albeit at the cost of developing a more complex interface. Such issues should be presented at PDR, along with a suitable discussion of AFS/DFS features aimed at eliminating some of the WAN-based NFS difficulties.

- **Flexible layered protocol using TCP/IP and SLIP:** TCP/IP useful for direct connection of microcomputers to network. SLIP useful for serial connections and MODEMs. Both fit as part of a layered communications protocol (p. 67).

Categorization: PDR, EVO (S/L)

This is really a non-issue, as there are a number of solutions (including COTS) available which provide complete solutions for Ethernet or serial line-based connections. Microsoft, for example, now ships TCP/IP and both Ethernet and SLIP/PPP drivers as part of its standard operating system. Trends towards "Internet-ready" Unix workstations are expected to cause Unix vendors to follow suit.

- **Inter-DAAC connectivity evolve to ATM and bandwidth on demand:** In the context of a small number of DAACs with dedicated communications links, expectation is that upgrades will be made as technology evolves from current T-1 or fractional T-1 to support ATM and bandwidth on demand (p. 86). This is recommended as the short and long term approach.

Categorization: ECS, PDR, EVO (S/L)

These items are concerns for ESDIS in planning the network connections among the DAACs, and between the DAACs and the SCFs. The ECS architecture needs to be able to support bandwidth "reservations" for appropriate applications and/or users, and to leverage the relevant networking capabilities when they mature.

- **Telecomm. choices for non-science users diverse and heterogeneous CATV Coax (Analog and Digital)**: High penetration of CATV among non-science users makes it a logical choice for near term (p. 89).

Unshielded Twisted Pair (ADSL), Passive Optical Network: Long term possible comm technology with higher bandwidth. Assumes fiber to the home.

Hybrid Fiber and Wireless: High bandwidth, probably as part of the NII.

Categorization: EVO (S/L)

Again, these are issues for ESDIS in determining the optimal technologies to implement the bandwidth requirements levied by current distribution requirements. As these technologies mature and become "interwoven" into the fabric of the NII, the system should support their use in a transparent manner through common transport level protocols.

- **Network performance of ATM indicates 16% utilization (real testbeds):** End-to-end application performance of networks is dependent on many variables including the host configuration and software. The UND team recommendations are that ECS be involved in the actual use of these network systems in order to better predict performance. Several example network technologies are discussed in the context of specific configurations that may or may not be useful to ECS. In most cases, the effective utilization is far below the anticipated maximum throughput, which has important implications for overall architecture design (p. 99).

Categorization: ECS, PDR, R&D

ESDIS and the ECS team are aware of such studies, and are actively involved in a number of ATM testbedding efforts to better understand the issues and implications. They also need to track (and perhaps promote) advances in protocols for ATM.

- **BONeS simulation of inter-DAAC topology indicates hypercube, mesh fully interconnected outperform Version 0 topology:** A BONeS simulation compared the cell throughput and cell delay for three alternatives (hypercube, mesh, fully connected) to the Version 0 topology of the DAACs. In the case of throughput, as the traffic load was increased, the BONeS simulation correctly modeled the Version 0 topology and indicated that in the three alternatives, with additional links the bandwidth demand on a given link is lower since traffic is more efficiently routed through the other links, resulting in overall improved network efficiency and throughput. Fully connected provides the best throughput with mesh and hypercube slightly less efficient although significantly more efficient than Version 0. Cell delay was roughly constant for each of the four topologies as the traffic intensity increased. However, cell delay with respect to Version 0 was 25% better for hypercube, 33% better for mesh, and 38% better for fully connected (p. 128).

Categorization: PDR

These studies may provide useful tools to the network design team in determining the optimal layout for the inter-DAAC topology over time. The analyses in the UND studies need to be revised to take into account version 1 technology, bandwidth, and topology plans.

- **Optically switched WDM Networks provide virtual topology suited to application needs:** In the long term, a virtual topology of any kind (hypercube, mesh, fully) can be embedded over an irregular physical topology consisting of optical fiber links employing WDM (p. 131).

Categorization: EVO (L), R&D

Such techniques should be tracked as part of the long term networking plan.

- **Premature to recommend complex object management approaches**

such as CORBA: A decision to base ECS architecture on CORBA network services is premature and could be an impediment to use in the short term(especially by PARCs). Treat CORBA as a long term option for future ECS development (p. 47).

Categorization: PDR, EVO (S/L)

In the long term CORBA will provide the network services required by the open, distributed and extensible architecture of ECS. However, CORBA implementation in the early releases of ECS is not advisable due to the relative immaturity of object oriented COTS products for network services. Early versions of ECS will preserve the option to implement CORBA in the long term without making the system dependent upon the availability of CORBA COTS products in the short term. Assessment of the maturity of CORBA technologies will be an ongoing activity within the ECS development project.

4.3.3 Other Architectural Issues

- **Direct broadcast of data should be supported:** Direct broadcast of data is critical to providing widespread availability of data, especially for application needs with short turn around time (i.e., less than 48 hours).

Categorization: PRG

Programmatic issue that needs to be considered by ESDIS and NASA HQ.

- **Advisory panel should include non-science users.**

  Categorization: PRG

Programmatic issue that needs to be considered by ESDIS and NASA HQ.

- **Over-arching data policy is free data at near real-time**: Free data at least at the level 0 and 1, possibly higher levels. Recommendation based on the view that taxpayers have already paid for the data at some low level (not as value is added).

  Categorization: PRG

Current EOS policy is to distribute data at the marginal cost of fulfilling a user request. Changes to this policy fall within the purview of NASA HQ.

- **Defer selection of data products as long as possible:** This maximizes the options and potentially lowers the cost.

  Categorization: PRG

Specification of the products that will be produced and archived within ECS is the responsibility of ESDIS and NASA HQ.

- **EOSDIS provide educational outreach to engage millions:** Recommend better outreach to ensure EOSDIS success with the public in both short and long term. UND team has made a good start at providing this through several means including Mosaic home pages.

Public awareness and positive perception

Tutorials and courses

New multimedia tools, MTPE and PARC Mosaic pages,

webcrawlers/worms, AgPARC Mosaic page example

  Categorization: EVO (S/L), PRG

These are excellent suggestions which are implemented at the programmatic level within NASA via the use of Cooperative Agreement Notice programs and other outreach efforts.

- **Technology Assessment:** Consider assessments of both networks and computing trends.

  Categorization: ECS, PDR

This is being done through a number of channels, including an ECS Technology Assessment team, ECS and ESDIS prototyping efforts, NRAs, and CANs.

- **Scenario Assessment:** Several moderately detailed scenarios should be further assessed for relevance to ECS (even as it is currently focused).

  Categorization: ECS, PDR

The UND work includes some interesting additional use scenarios that should be considered in the ECS user modeling effort.

• **Data from DAACs is replicated to PARCs that need it:** Raw EOSDIS data and some higher products replicated from DAAC to PARC as needed by particular application area served by PARC. Place standing orders for data copies from DAACs (p. 42).

Categorization: EVO (S/L), PRG

This is currently supported within the current ECS architecture, commensurate with adequate resource availability for generation and distribution.

• **Store data encoded, decode on demand only:** Recommendation in the context of existing RWIC system to avoid spending time decoding observations—decode on demand only (p. 70).

Categorization: EVO (S/L)

This relates to the "precomputed products" vs. "compute-on-demand" tradeoffs to be made within the production subsystem of the current architecture.

• **DAACs archive and provide access for earth science users, and have large storage capability:** UND team view is that DAACs will be busy with archiving and serving SCFs and earth scientists and will not be able to support millions of non-science users. Therefore, they should concentrate on archive and science user support.

Categorization: ECS, PDR

This is precisely the rationale behind the value added provider (VAP) concept in the current architecture. The VAP network will serve as a "secondary provider" layer to a variety of "vertical" data markets, including additional earth scientists, policy makers, and the educational community.

Appendix A. IAS Review Panel Members

Bruce Barkstrom	LaRC
Bud Booth	NOAA/SAAC
Dixon Butler, Chairman	NASA HQ
John Dalton	GSFC
David DeWitt	Univ of Wisc
Mark Elkington	ECS - HAIS
G. Dave Emmitt	Simpson Weather Assoc., Inc
Robert Evans	Univ of Miami
Steve Fox	HAIS
David Glover	WHOI
Joe Guzek	HAIS
Greg Hunolt	GSFC
Ed Lerner	HAIS
Robert Mairs	NOAA SAAC
Gail McConaughy	GSFC
Roberta Miller	CIESIN
Mark Settle	HAIS
Martha (Marti) Szczur	GSFC
R. J. Thompson	EDC
Steve Wharton	GSFC

This page intentionally left blank.